What is claimed is:

1. A method comprising:

receiving a request from a first computer system for identification of a second computer system;

retrieving an identifier that identifies the second computer system;

encrypting the identifier with a key associated with the first computer system to produce a hash value; and

providing the hash value to the first computer system in response to the request.

2. The method of claim 1, wherein the act of retrieving the identifier comprises:

retrieving a processor number that identifies a processor of the second computer system.

3. The method of claim 2, further comprising:

executing a processor instruction; and

retrieving the number in response to the execution of the instruction.

4. The method of claim 1, further comprising:

receiving the key from the first computer system.

5. The method of claim 1, wherein the key indicates an address of a web site.

6. A computer system comprising:

an interface adapted to:

receive a request from another computer system for identification of the first computer system, and

furnish a hash value that identifies the first computer system to said another computer system; and

a processor coupled to the interface and adapted to:

encrypt an identifier that identifies the first computer system with a key associated with said another computer system to produce the hash value.

7. The computer system of claim 6, wherein the identifier comprises a processor number that identifies the processor.

8. The computer system of claim 6, wherein the processor comprises:

a memory adapted to store microcode for performing the encryption; and

a control unit coupled to the memory and adapted to execute the microcode to perform the encryption.

9. The computer system of claim 6, wherein the processor is further adapted to:

interact with the interface to receive the key from said another computer system.

10. An article comprising a storage medium readable by a first processor-based system, the storage medium storing instructions to cause a processor to:

receive a key from another processor-based system for identifying the first system,

determine whether the key is valid,

based on the identification, selectively authorize encryption of an identifier that identifies the first system with the key to produce a hash value.

11. The article of claim 10, the storage medium storing instructions to cause the processor to:

use an address of said another system to determine whether the key is valid.

12. The article of claim 11, wherein the key indicates an URL address.

13. The article of claim 10, the storage medium storing instructions to cause the processor to:

execute an instruction to cause the processor to subsequently use the key to produce the hash value.

14. The article of claim 10, wherein the identifier comprises a processor number.

15. A microprocessor comprising:

an instruction unit adapted to indicate when the instruction unit receives an instruction that requests an identifier that identifies the microprocessor;

an execution unit coupled to the instruction unit and adapted to, in response to the indication from the instruction unit, encrypt a key with the identifier to produce a hash value; and

a bus interface unit coupled to the execution unit and adapted to furnish an indication of the hash value to external pins of the microprocessor.

16. The microprocessor of claim 15, wherein the execution unit comprises:

a control unit coupled to the algorithmic unit and the registers; and

a memory coupled to the control unit and storing microcode to cause the control unit to use the key and the identifier to produce the hash value.

17. The microprocessor of claim 15, wherein the identifier comprises a processor number.

18. The microprocessor of claim 15, wherein the execution unit is adapted to use a one way hash function to produce the hash value.

19. The microprocessor claim 15, wherein the execution unit is adapted to use a non-commutative hash function to produce the hash value.

20. The microprocessor of claim 15, wherein the execution unit is adapted to use a collision free hash function to produce the hash value.